

3772 = 38
Your name: _____ Your student number: 930772

Department of Computer Science
University of Saskatchewan
CMPT 370-03

Midterm Examination
October 31, 2002

Time Limit: 75 minutes

Total Marks: 50

This is a closed book exam. Please write your answers legibly in the space provided on the examination paper. In the discussion questions you may use point form as long as your answer is coherent. If you need more space, use the back of the page. Rough work can be done in the answer booklets. Be sure to budget your time appropriately so you can answer all questions. The number of marks assigned to each question is a rough guide as to the relative amount of time to spend on that question. Good luck.

15 **Section 1: Short Discussion [3 marks for each question; total for the section: 21]**

Each of the questions in this section require at most a few sentences to answer.

3 1. What is the difference between a domain model and a design class diagram in the unified process (UP)?

Domain model describes the real world environment and a design class diagram shows the real classes that will be designed for the software. Also, the design class diagram shows visibility and the domain model does not.

2 2. What is visibility? Why is it important to design?

Visibility is what objects can "see" other objects. If too many objects are visible to other objects it leads to high coupling in the system which is undesirable.

3 3. What is the difference between debugging and testing?

Debugging is checking to see if the code is correct and running and testing is to see if the system does what you want it to.

2 4. What is the system boundary? Why is it important to know where this is?

The system boundary is where messages are passed from outside/inside the user to the actual system. It gives you a representation of the real world objects and the actual software.

5. What are the advantages of the iterative prototyping methodology (promoted by the UP) when compared to the waterfall model?

2 The advantages are you are always refining things instead of going with one way in the waterfall model where if you find a problem you have to go back to the start. The iterative prototyping method leads to more clearly defined due dates and a quicker release of software.

6. It has been said that "the best object-oriented designs don't tend to resort to simulated people, but rather tend to encourage animated objects." Explain what this means with an example from the point-of-sale system used in the textbook.

3 This means that a person object should not be doing all the work instead another object should be taking care of it. An example is the payment class taking care of credit card payment instead of a person.

7. What is a "layered architecture"? Why is it useful to "architect" a system this way?

0 A layered architecture is a system that builds upon itself until it is finished. It is useful because it allows for single things to be done quickly and then add on the harder designs.

Section 2 - Design Patterns [total marks for the section: 5]

The answers in this section should be very short, just a few words each.

8. In each of the following scenarios a GRASP design pattern has either been properly used or violated. In each case, first indicate which type of use the scenario represents: a proper use or a violation. Then, indicate which design pattern has been properly used or violated (restrict your answer to the single most appropriate GRASP pattern). [5 marks]

(a) Each object in a system has been assigned a single method.

1 Violation. High Cohesion.

(b) An object is updated in different ways depending on circumstances, so a class representing the object is given different specialized sub-classes for each type of circumstance.

1 Proper Use. Polymorphism.

(c) A single object in a system must interact with a vast multitude of other objects.

1 Violation. Low Coupling.

(d) In the point of sale system in the text book the **Payment** object is assigned the responsibility of getting credit card information from the customer.

1/2 Proper Use. Expert.
Controller.

(e) In the student record system designed in class, the **Professor** object adds a new section to a course.

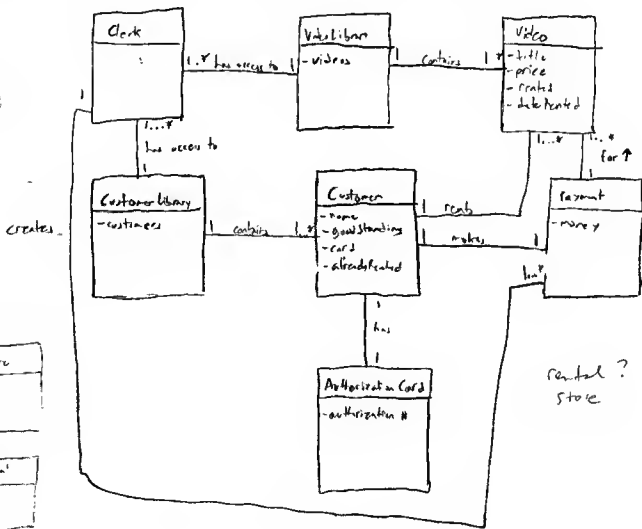
0 Proper Use. Creator.
Violation.

Section 3 – Analysis and Design [marks for each part indicated; total for the section: 24]

This question has many parts. Do your best to answer each part in the space available.

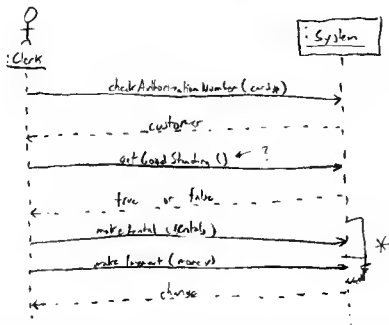
9. The GrandView antique video rental store is automating its entire operation and you are to design the system. The GrandView store works much like any video store. Customers come into the store, browse through shelves full of videos to find the (old) movies they are interested in, take the videos of their choice to a clerk, and present their GrandView authorization card (giving them the right to rent videos from GrandView). Using this card, the clerk then checks that the customer is in "good standing" with the store and thus is allowed to rent the videos. The clerk then totals up the cost of the video rentals, takes payment from the customer (GrandView, as befits an antique store, is a cash only operation!), and returns the customer's change, if any. The customer then leaves with the videos. [24 marks]

- (a) Draw a domain model for this situation. Show the important objects, their main associations, the cardinalities (multiplicities), and a few important attributes. [8 marks]



Question 9 (continued)

- (b) Draw a system sequence diagram showing the main interactions between the outside world and the system. [4 marks]



- (c) Assume there is a system operation **makeRental** that has been identified during analysis. **makeRental** is carried out to initiate the rental of one or more videos. Write a software contract for **makeRental** showing in particular the pre-conditions, and the post-conditions and indicating for each post-condition the kind of post-condition it is (i.e. association formed or deleted, instance created or deleted, attribute modified). [4 marks]

makeRental (rentals, CUSTOMER-CLASS)

preconditions: rentals is not empty

customer exists

customer is in good standing

→ is this an object?

new rental object?
new rental form?

postconditions: (movies) were associated with customer (Association formed)

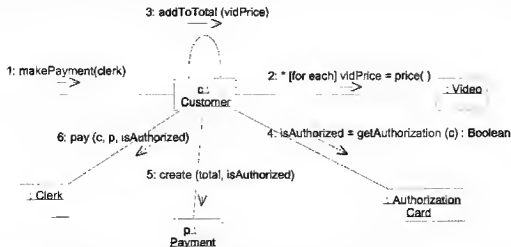
movie's rental attribute were changed to (false) (attribute modified)

movie's due rental were changed to new date (attribute modified)

movies added to customer's alreadyRented variable (attribute modified)

Question 9 (continued)

- (d) Assume there is another system operation **makePayment** that has also been identified. **makePayment** computes the total amount it will cost to rent the various videos the customer has selected so the clerk can tell the customer the total cost. The following collaboration diagram is meant to represent the **makePayment** operation.



There are many serious problems with the *design* represented in this collaboration diagram. In a sentence or two each (but no more) describe *four* such *design* problems and indicate what principles are being violated in each case. [8 marks]

Clerk would make a better controller than customer and so make payment should have in a customer. This is a bad design of a Controller which led to high coupling.

Determining whether the customer is authorized to rent videos should be done first before the price is determined. This will lead to less work being done if the customer is not authorized.

Clerk should take care of getting the total price for the videos because the clerk knows about them. This is a bad use of Information Expert.

Clerk should also create the payment class because it knows the information to do so. This is a bad use of the Controller pattern.

Trick or Treat!